DW_fp_recip

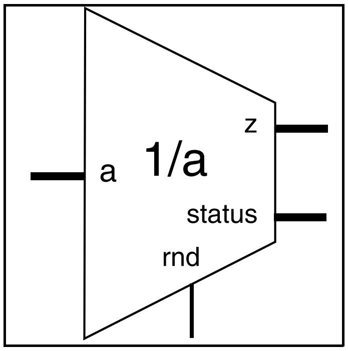
Floating-Point Reciprocal

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

Revision History

- The precision format is parameterizable for either IEEE single, double precision, or a user-defined custom format
- Hardware for denormal numbers of IEEE 754 standard is selectively provided
- Both IEEE 754 standard rounding modes and the faithful rounding with 1 ulp error are supported
- Accuracy conforms to IEEE 754 Floating-point standard



Description

DW_fp_recip is a floating-point reciprocal unit that calculates `z` = 1/`a` where `a` is a floating-point value. DW_fp_recip supports the IEEE 754 compatible rounding modes as well as the faithful rounding that admits maximum 1 ulp error.

Component pins are described in Table 1-1 and configuration parameters are described in Table 1-2.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
a	(*exp_width* + *sig_width* + 1) bits	Input	Divisor
rnd	3 bits	Input	Rounding mode; supports all rounding modes described in the *Datapath Floating-Point Overview*; The `rnd` port takes effect only when *faithful_round* = 0.
z	(*exp_width* + *sig_width* + 1) bits	Output	Quotient of 1/a
status	8 bits	Output	Status flags for the result `z` For details, see STATUS Flags in the *Datapath Floating-Point Overview.*

Table 1-2 Parameter Description

Parameter	Values	Description
sig_width	3 to 60 bits Default: 23	Word length of fraction field of floating-point numbers `a`, and `z`

Table 1-2 Parameter Description (Continued)

Parameter	Values	Description
exp_width	3 to 31 bits Default: 8	Word length of biased exponent of floating-point numbers `a`, and `z`
ieee_compliance	0 or 1 Default: 0	Level of support for IEEE 754: ■ 0: No support for IEEE 754 NaNs and denormals; NaNs are considered infinities and denormals are considered zeros ■ 1: Fully compliant with the IEEE 754 standard, including support for NaNs and denormals For more, see IEEE 754 Compatibility in the *Datapath Floating-Point Overview*.
faithful_round	0 or 1 Default: 0	Choose either a specific rounding mode (set by `rnd`) or a general rounding mode that allows maximum 1 ulp error ■ 0: Rounding mode is specific, as set by the `rnd` port; this choice increases the size of the resulting implementation. ■ 1: Rounding mode is general and, for *sig_width* ≤ 28, allows a maximum of 1 ulp error; this choice decreases the size of the resulting implementation[a]. When *faithful_round* = 1, note the following: - The inexact status flag in the output is not meaningful. - The other status flags will match one of the possible outputs for the calculation when *faithful_round* = 0.

[a] When *faithful_round* = 1 and *sig_width* > 28, the result can exceed 1 ulp for some corner cases. Configurations tested for this parameter range do not show errors larger than 2 ulps.

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
rtl	Synthesis model	DesignWare

Table 1-4 Simulation Models

Model	Function
DW02.DW_FP_RECIP_CFG_SIM	Design unit name for VHDL simulation
dw/dw02/src/DW_fp_recip_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW_fp_recip.v	Verilog simulation model source code

DW_fp_recip provides the hardware for denormal numbers and NaNs of IEEE 754 standard. If the parameter *ieee_compliance* = 0, denormal numbers are considered as zeros and NaNs are considered as infinity. Otherwise, denormal numbers and NaNs become effective and additional hardware to manipulate denormal numbers is integrated.

For more information about the floating-point system defined for all the DW floating-point components, including `status` flag bits, and integer and floating-point formats, refer to the *Datapath Floating-Point Overview*.

Alternative Implementation of Floating-point Reciprocal using DW_lp_fp_multifunc

The floating-point reciprocal operation can also be implemented by DW_lp_fp_multifunc component, which evaluates the value of floating-point reciprocal with 1 ulp error bound. There will be 1 ulp difference between the value from DW_lp_fp_multifunc and the value from DW_fp_recip. Performance and area of the synthesis results are different between the DW_fp_recip and reciprocal implementation of the DW_lp_fp_multifunc, depending on synthesis constraints, library cells and synthesis environments. By comparing performance and area between the reciprocal implementation of DW_lp_fp_multifunc and DW_fp_recip component, the DW_lp_fp_multifunc provides more choices for the better synthesis results. Below is an example of the Verilog description for the floating-point reciprocal of the DW_lp_fp_multifunc. For more detailed information, see the DW_lp_fp_multifunc datasheet.

```
DW_lp_fp_multifunc #(sig_width, exp_width, ieee_compliance, 1, faithful_round, 1) U1 (
    .a(a),
    .func(16'h0001),
    .rnd(rnd),
    .z(z),
    .status(status)
);
```

Suppressing Warning Messages During Verilog Simulation

The Verilog simulation model includes macros that allow you to suppress warning messages during simulation.

To suppress all warning messages for all DWBB components, define the DW_SUPPRESS_WARN macro in either of the following ways:

- Specify the Verilog preprocessing macro in Verilog code:

  ```
  `define DW_SUPPRESS_WARN
  ```

- Or, include a command line option to the simulator, such as:

  `+define+DW_SUPPRESS_WARN` (which is used for the Synopsys VCS simulator)

The warning messages for this model include the following:

- If an invalid rounding mode has been detected on `rnd`, the following message is displayed:

  ```
  WARNING: <instance_path>:
      at time = <timestamp>: Illegal rounding mode.
  ```

  To suppress this message, use the DW_SUPPRESS_WARN macro explained earlier.

Related Topics

- Datapath Floating-Point Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_Foundation_comp_arith.all;

entity DW_fp_recip_inst is
      generic (
        inst_sig_width : POSITIVE := 23;
        inst_exp_width : POSITIVE := 8;
        inst_ieee_compliance : INTEGER := 0;
        inst_faithful_round : INTEGER := 0
        );
      port (
        inst_a : in std_logic_vector(inst_sig_width+inst_exp_width downto 0);
        inst_rnd : in std_logic_vector(2 downto 0);
        z_inst : out std_logic_vector(inst_sig_width+inst_exp_width downto 0);
        status_inst : out std_logic_vector(7 downto 0)
        );
    end DW_fp_recip_inst;


architecture inst of DW_fp_recip_inst is

begin

    -- Instance of DW_fp_recip
    U1 : DW_fp_recip
    generic map (
          sig_width => inst_sig_width,
          exp_width => inst_exp_width,
          ieee_compliance => inst_ieee_compliance,
          faithful_round => inst_faithful_round
          )
    port map (
          a => inst_a,
          rnd => inst_rnd,
          z => z_inst,
          status => status_inst
          );


end inst;
```

HDL Usage Through Component Instantiation - Verilog

```
module DW_fp_recip_inst( inst_a, inst_rnd, z_inst, status_inst );

parameter inst_sig_width = 23;
parameter inst_exp_width = 8;
parameter inst_ieee_compliance = 0;
parameter inst_faithful_round = 0;


input [inst_sig_width+inst_exp_width : 0] inst_a;
input [2 : 0] inst_rnd;
output [inst_sig_width+inst_exp_width : 0] z_inst;
output [7 : 0] status_inst;

    // Instance of DW_fp_recip
    DW_fp_recip #(inst_sig_width, inst_exp_width, inst_ieee_compliance,
inst_faithful_round) U1 (
                .a(inst_a),
                .rnd(inst_rnd),
                .z(z_inst),
                .status(status_inst) );

endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
July 2020	DWBB_201912.5	■ Adjusted the description of the *ieee_compliance* parameter in Table 1-2 on page 1 ■ Added "Suppressing Warning Messages During Verilog Simulation" on page 3
April 2020	DWBB_201912.3	■ Updated the value range for *sig_width* in Table 1-2 on page 1 ■ Added default values for *sig_width* and *exp_width* in Table 1-2 on page 1 ■ Updated the description of `rnd` in Table 1-1 on page 1 and *faithful_round* in Table 1-2 on page 1 ■ For STAR 3124623, added note to Table 1-2 on page 1 to update the error range when *faithful_round* = 1. This update is based on a limitation found during the investigation of STAR 3124623. Also, Table 1-5, "Error Ranges (ε = z – a/b)" was removed.
January 2020	DWBB_201912.1	■ Corrected port names for DW_lp_fp_multifunc in "Alternative Implementation of Floating-point Reciprocal using DW_lp_fp_multifunc" on page 3
June 14, 2019	DWBB_201903.2	■ Removed reference to minPower library in "Alternative Implementation of Floating-point Reciprocal using DW_lp_fp_multifunc" on page 3 ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com